

Course Title	Software Engineering Professional Placement				
Course Code	ACSC390				
Course Type	BSc Computer Science: Required Course				
Level	BSc (Level 1)				
Year / Semester	3rd year / 6th semester and Summer Term				
Teacher's Name	Dr. Achilleas Achilleos				
ECTS	6	Lectures / week	2	Laboratories/week	0
Course Purpose	<p>The aim of this course is to introduce students to the professional practice of software engineering. It focuses on the development of modern applications and services due to their paramount importance for students, in terms of acquiring practical software engineering and communication skills. The adopted software engineering model for this course is agile development. Based on that, the students will have direct contact with the clients (i.e., companies or research groups) to gather and analyse requirements to design and implement the software application and services required in incremental steps. A Minimum Viable Product (MVP), i.e., a prototype, is created and then tested and improved over multiple cycles, until it becomes the final product. Agile development allows receiving feedback continuously to respond faster to changes, so as to gradually deliver a higher-quality software product. The course is linked to a student placement at a company or working with a research group over the summer term with 160 hours that will be recorded in the logbook provided to the student.</p>				
Learning Outcomes	<p>Upon successful completion of the course students will be able to:</p> <ul style="list-style-type: none"> • Understand the importance of project management through the collaboration with the company's or research group's team. • Develop analytical, design, development and communication skills to engage with agile software development of real systems. • Recognize software reuse as an important parameter in building modern applications and services based on the concept of open source software development. • Apply in practice their knowledge on agile development for the incremental analysis, design and development of software application and services. • Extract requirements from the client, analyze and organize the information, formulate ideas and to communicate with the client. • Convert requirements into software design and implementation in incremental steps applying modern methods and techniques. • Build a software system according to a client's (company project or internal academic/research project) requirements and deadlines. 				

* The weekly schedule includes lectures, laboratories and meetings with the client.

Prerequisites	ACSC223, ACSC382, ACSC383.	Corequisites	None.
Course Content & Course Schedule	<p>1. Agile Development and Project Management – (1 Week)</p> <ul style="list-style-type: none"> - Agile Fundamentals. Plan-Driven and Agile Development. Management activities, Project planning, Project scheduling, Risk management. RedMine and Google Drive for Project Collaboration. GitHub Software Development Platform. (Lecture – 1 Week) <p>2. Domain Problem and Requirements – (3 Weeks)</p> <ul style="list-style-type: none"> - Minimum Viable Product (MVP), User Stories. Tasks. Focus on People not Software Processes: Surveys, Interviews, Focus Groups. (Lecture – 1 week) - Students will schedule and meet with client to perform the interviews, surveys and/or focus groups with the relevant stakeholders. (Meetings – 2 weeks) <p>3. Architecture Design and Software Technologies – (1 Week)</p> <ul style="list-style-type: none"> - Students will meet with clients for requirements validation and refinement, as well as deciding on the architecture definition, the software technologies and IDEs to be used. (Meeting – 1 week) <p>4. Design and Implementation – (4 Weeks)</p> <ul style="list-style-type: none"> - Students will work in defined teams at the laboratory with the support of the lecturer on the design and implementation of the Minimum Viable Product (MVP). (Laboratory – 4 weeks) <p>5. Debugging, User Experience and Usability Testing – (3 Weeks)</p> <ul style="list-style-type: none"> - User Experience and Usability Testing, User Experience Questionnaire (UEQ), System Usability Scale (SUS). Definition of the UEQ and SUS surveys (Lecture – 1 week) - Students will schedule and meet with client to perform the MVP testing and evaluation and receive feedback. The UEQ and SUS questionnaires will be used for the evaluation. (Meetings – 2 weeks) <p>6. Presentation and MVP prototype demo – (1 Week)</p> <ul style="list-style-type: none"> - Students will present the results of the project to the client and lecturer with the help of a PowerPoint presentation and a live demo of the MVP prototype. 		
Teaching Methodology	<p>The course is structured around lectures, laboratories and client meetings in order to provide the flexibility and agility required for the students to engage with software engineering at a practical level. The students at the beginning of the course will be divided in teams and they will select a software engineering project from the list of projects posted by companies and/or research groups of the department. The projects will be posted by company mentors and/or academics using the Department's projects portal. Details on the schedule are available in the previous section. The course is delivered to the students with the help of computer presentations, code examples where applicable and through interactive laboratory sessions, as well</p>		

* The weekly schedule includes lectures, laboratories and meetings with the client.

	<p>as scheduled meetings with the clients. The material is available on the e-learning system and is the main resource for students to use in their study, in combination with the recommended textbooks and references. The course continues over the summer period with the student placement at the company or research group, where the students have the chance to continue to work and receive frequent guidance and feedback for delivering the final product.</p> <p>The course is designed with flexibility and agility and a key focus on the professional practice of software engineering. This allows to help and motivate students to engage in solving real-world software engineering problems. Finally, the assessment of the course is based on the final presentation and the implementation of the MVP prototype (Spring semester), as well as the quality of the final product delivered at the end of the summer term.</p>
Bibliography	<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Software Engineering, 10th Edition, by Ian Sommerville (Author), 816 pages, Publisher: Pearson; 10 edition (April 3, 2015), Language: English, ISBN-10: 0133943038, ISBN-13: 978-0133943030. <p>References:</p> <ol style="list-style-type: none"> 1. John Vlissides, Ralph Johnson, Richard Helm, Erich Gamma, "Design Patterns: Elements of Reusable Object-Oriented Software", Publisher: Addison-Wesley Professional, Release Date: October 1994, ISBN: 0201633612 2. M. Seidl, M. Scholz, C. Huemer, G. Kappel, "UML@Classroom: An Introduction to Object-Oriented Modeling" (Undergraduate Topics in Computer Science) 2015 Edition, Online Available: https://link.springer.com/content/pdf/10.1007%2F978-3-319-12742-2.pdf 3. Textbook Resources: https://iansommerville.com/software-engineering-book/. 4. UML.org Resources page – https://www.uml.org/resource-hub.htm. 5. Homepage of ArgoUML – http://argouml.tigris.org/.
Assessment	<ul style="list-style-type: none"> • Presentation (team – 5%), (individual – 5%) 10% • MVP Product (team – 30%), (individual – 10%) 40% • Final Product (team – 40%), (individual – 10%) 50%
Language	English.

* The weekly schedule includes lectures, laboratories and meetings with the client.